Project: Weather Analysis

Data Preparation and Analysis Documentation

Data Preparation

1. Introduction

This document outlines the steps taken to clean and preprocess a weather dataset for analysis.The dataset was obtained from Kaggle and contains information about temperature, humidity, rainfall, and other weather-related metrics.

2. Data Loading

In [ ]:
```
df = pd.read_csv('weather.csv')
df.head()
```

3. Data Exploration

Initial exploration steps were conducted to understand the structure and content of the dataset.

In [ ]:
```
# Display the first few rows of the dataset
print(weather_data.head())

# Get an overview of the dataset
print(weather_data.info())

# Statistical summary of the dataset
print(weather_data.describe())
```

4. Handling Missing Values

Missing values were identified and imputed with the mean value for the respective columns.

In [ ]:
```
# Check for missing values
print(weather_data.isnull().sum())

# Impute missing values (example: filling with mean)
weather_data['temperature'].fillna(weather_data['temperature'].mean(), inplace
weather_data['humidity'].fillna(weather_data['humidity'].mean(), inplace=True)
```

5. Handling Outliers

Outliers were identified using box plots and removed using the interquartile range (IQR) method.

In [ ]:
```
# Visualize outliers using box plots
sns.boxplot(x=weather_data['temperature'])
plt.show()

# Define a function to remove outliers using IQR
def remove_outliers(column):
    Q1 = weather_data[column].quantile(0.25)
    Q3 = weather_data[column].quantile(0.75)
    IQR = Q3 - Q1
    return weather_data[(weather_data[column] >= Q1 - 1.5 * IQR) & (weather_da

# Remove outliers for the 'temperature' column
weather_data = remove_outliers('temperature')
```

6. Handling Other Inconsistencies

Additional data cleaning steps included converting the 'date' column to datetime format and fixing errors in the 'humidity' column.

In [ ]:
```
# Example: Convert 'date' column to datetime format
weather_data['date'] = pd.to_datetime(weather_data['date'])

# Example: Fix errors in the 'humidity' column
weather_data['humidity'] = weather_data['humidity'].apply(lambda x: min(100, x
```

In [ ]:
```
```

Advanced Analysis

1. Introduction

This section documents the advanced analysis performed on the cleaned weather dataset, aiming to derive insights and patterns from the data.

2. Analysis Steps

Various visualizations and calculations were employed to conduct the advanced analysis.

In [ ]:
```
# Example: Line chart for temperature trends
sns.lineplot(x='date', y='temperature', data=weather_data)
plt.title('Temperature Trends Over Time')
plt.show()
```

3. Key Visualizations

Key visualizations were created to highlight specific insights from the weather data.

In [ ]:
```
# Example: Scatter plot for humidity vs. temperature
sns.scatterplot(x='temperature', y='humidity', data=weather_data)
plt.title('Humidity vs. Temperature')
plt.show()
```

4. Insights Derived

Insights were derived from the visualizations and analyses, providing valuable information about weather patterns.

Temperature tends to rise during summer months. There is a negative correlation between temperature and humidity. Monthly rainfall shows a peak during the monsoon season.

5. Conclusion

In conclusion, the data preparation and advanced analysis have revealed significant insights into the weather dataset. These findings can be used for further exploration and decision-making in relevant domains.